**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: THỰC TẬP CƠ SỞ**

**MÃ HỌC PHẦN: INT13147**

**ĐỀ TÀI: Xây dựng website tìm kiếm việc làm**

**Giảng viên hướng dẫn : Ths.Bùi Văn Kiên**

**Nhóm : 10**

**Sinh viên thực hiện : Đỗ Gia Phong - B22DCCN613**

**HÀ NỘI 2025**

**MỤC LỤC**

**MỞ ĐẦU**

Trong bối cảnh nền kinh tế đang không ngừng phát triển và chuyển mình theo xu hướng công nghệ hóa, số hóa, nhu cầu tìm kiếm việc làm của người lao động cũng như nhu cầu tuyển dụng của các doanh nghiệp ngày càng tăng cao và trở nên đa dạng. Tuy nhiên, việc kết nối giữa người tìm việc và nhà tuyển dụng vẫn còn tồn tại nhiều khó khăn, đặc biệt là ở những khu vực chưa được tiếp cận đầy đủ với các kênh thông tin hiện đại. Các phương pháp truyền thống như phát tờ rơi, thông báo trực tiếp hay đăng tin trên báo giấy đang dần trở nên lỗi thời, không còn đáp ứng được yêu cầu về tốc độ, tính chính xác và khả năng cập nhật liên tục của thị trường lao động hiện đại.

Trong những năm gần đây, sự phổ biến của internet và các nền tảng web đã tạo điều kiện thuận lợi cho việc phát triển các hệ thống hỗ trợ tuyển dụng và tìm kiếm việc làm trực tuyến. Thông qua các trang web, người dùng có thể dễ dàng tạo hồ sơ cá nhân, tìm kiếm công việc phù hợp theo ngành nghề, vị trí địa lý, mức lương, kỹ năng yêu cầu và nhiều tiêu chí khác. Đồng thời, các nhà tuyển dụng cũng có thể nhanh chóng tiếp cận nguồn ứng viên tiềm năng, đăng tin tuyển dụng và quản lý hồ sơ ứng tuyển một cách hiệu quả.

Dựa trên thực tế đó, việc xây dựng một **website tìm kiếm việc làm** đóng vai trò vô cùng quan trọng và cấp thiết, nhằm tạo ra một cầu nối hiệu quả giữa người lao động và doanh nghiệp. Hệ thống không chỉ giúp giảm thiểu chi phí và thời gian cho cả hai bên mà còn tạo điều kiện cho người tìm việc tiếp cận được nhiều cơ hội việc làm phù hợp, đồng thời hỗ trợ doanh nghiệp nâng cao hiệu quả trong quá trình tuyển dụng.

Đề tài “Xây dựng website tìm kiếm việc làm” được thực hiện nhằm mục tiêu tạo ra một nền tảng trực tuyến thân thiện, dễ sử dụng, tích hợp đầy đủ các tính năng cần thiết như tìm kiếm công việc theo tiêu chí, đăng ký tài khoản, tạo hồ sơ cá nhân, ứng tuyển trực tuyến, quản lý tin tuyển dụng,... Bên cạnh đó, hệ thống cũng đảm bảo tính bảo mật thông tin người dùng và khả năng mở rộng trong tương lai để phù hợp với sự phát triển của thị trường lao động.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Giới thiệu hệ thống

Hệ thống tìm kiếm việc làm được thiết kế nhằm giúp ứng viên tìm được công việc phù hợp, hỗ trợ nhà tuyển dụng quản lý tuyển dụng hiệu quả và cung cấp công cụ giám sát, phân tích cho quản trị viên. Hệ thống bao gồm **ba vai trò chính: User (Ứng viên), Recruiter (Nhà tuyển dụng) và Admin (Quản trị viên).**

## Các thành phần chính của hệ thống

### Phân hệ User (Người dùng)

Ứng viên là người dùng chính của hệ thống, có thể :

**Xác thực tài khoản :**

* Đăng ký tài khoản bằng username/password hoặc Google OAuth.
* Đăng nhập, đăng xuất, lấy lại mật khẩu khi quên

**Tìm kiếm và quản lý công việc :**

* Tìm kiếm việc làm theo từ khóa, ngành nghề, địa điểm.
* **Xem chi tiết công việc**, bao gồm mô tả, yêu cầu, lương, địa điểm, công ty đăng tuyển.
* **Ứng tuyển công việc**, theo dõi trạng thái ứng tuyển (**thành công, bị từ chối**).
* Quản lý danh sách **công việc yêu thích**.

**Cá nhân hóa thông tin**

* Chỉnh sửa hồ sơ cá nhân, cập nhật kỹ năng, kinh nghiệm.

**Nhận thông báo :**

* Xem danh sách thông báo hệ thống.
* Nhận **thông báo đẩy** về việc tuyển dụng, trạng thái ứng tuyển.

### Phân hệ Recruiter (Nhà tuyển dụng)

Nhà tuyển dụng có thể:

**Quản lý tin tuyển dụng**:

* Thêm, chỉnh sửa, xóa **công việc tuyển dụng**.
* Xem danh sách công việc đang tuyển.

**Quản lý ứng viên**:

* Xem danh sách ứng viên đã ứng tuyển.
* **Phê duyệt hoặc từ chối** ứng viên.

**Cập nhật thông tin công ty**:

* Chỉnh sửa thông tin công ty hiển thị trên hệ thống.

### Phân hệ Admin (Quản trị viên)

Quản trị viên chịu trách nhiệm giám sát và vận hành hệ thống:

**Quản lý người dùng:**

* Xem danh sách ứng viên, nhà tuyển dụng.
* Xóa tài khoản vi phạm.

**Quản lý công việc**:

* Giám sát danh sách công việc đang được tuyển dụng.

**Phân tích & báo cáo**:

* Xem **thống kê theo ngành nghề, số lượng ứng viên, tỉ lệ ứng tuyển thành công**.

## Kiến trúc hệ thống

### Kiến trúc tổng quan

Hệ thống được xây dựng theo mô hình **Monolithic** , với các thành phần chính:

**FrontEnd :**

* Sử dụng **HTML, CSS, JS, Bootstrap** để xây dựng giao diện.
* Kết nối với API từ backend để hiển thị dữ liệu theo thời gian thực.

**Backend :**

* Xây dựng bằng **Spring Boot (Java)**
* Cung cấp API RESTful cho frontend.
* Tích hợp AI để phân tích dữ liệu ứng viên.

**Database :**

* **PostgreSQL** để lưu trữ thông tin ứng viên, công việc.

## Quy trình hoạt động của hệ thống

### Quy trình ứng viên tìm việc

* Đăng nhập / Đăng ký tài khoản.
* Tìm kiếm công việc theo ngành nghề, địa điểm, mức lương.
* Xem chi tiết công việc và nộp đơn ứng tuyển.
* Theo dõi trạng thái ứng tuyển (đang xét duyệt, thành công, bị từ chối).
* Nhận thông báo khi có cập nhật từ nhà tuyển dụng.

### Quy trình nhà tuyển dụng đăng tuyển

* Đăng nhập tài khoản nhà tuyển dụng.
* Thêm công việc mới, điền thông tin mô tả, yêu cầu, mức lương.
* Quản lý danh sách công việc đang tuyển.
* Xem danh sách ứng viên, phê duyệt hoặc từ chối.
* Cập nhật thông tin công ty để tăng độ tin cậy.

### Quy trình quản trị viên

* Quản lý người dùng, kiểm tra tài khoản vi phạm.
* Giám sát công việc.
* Xem báo cáo phân tích, hỗ trợ cải thiện trải nghiệm người dùng.

# MỘT SỐ CÔNG NGHỆ SỬ DỤNG

## Front End

### HTML (HyperText Markup Language)

* HTML ngôn ngữ đánh dấu dùng để xây dựng cấu trúc cơ bản của một trang web.
* Chức năng chính :

Định nghĩa các phần tử như: tiêu đề, đoạn văn, hình ảnh, biểu mẫu, nút bấm, bảng, liên kết, v.v.

Là "bộ xương" của giao diện người dùng, giúp trình duyệt hiểu được nội dung cần hiển thị.

### CSS (Cascading Style Sheets)

* CSS là ngôn ngữ dùng để tạo kiểu dáng, định dạng và làm đẹp cho trang web.
* Chức năng chính :

Quy định màu sắc, phông chữ, bố cục, độ rộng, khoảng cách, hiệu ứng chuyển động,...

Giúp giao diện trở nên trực quan và dễ sử dụng hơn.

### JS (Javascript)

* Ngôn ngữ lập trình phía client giúp xử lý logic, tương tác, và hành vi động của website.
* Chức năng chính :

Bắt sự kiện người dùng (click, nhập dữ liệu, cuộn trang,...)

Giao tiếp với backend (qua API)

Thay đổi nội dung trang web mà không cần tải lại (AJAX)

### Bootstrap 5

* Framework CSS & JS phổ biến giúp xây dựng giao diện web nhanh chóng, đẹp và tương thích đa thiết bị (responsive).
* Chức năng chính :

Bộ **component sẵn có**: navbar, button, form, card, modal,...

**Lưới 12 cột (Grid system):** dễ dàng bố trí giao diện responsive.

**Tùy chỉnh nhanh:** thông qua class mà không cần viết CSS nhiều.

* Ưu điểm :

Giảm thời gian phát triển.

Hỗ trợ tốt trên mọi trình duyệt.

Tương thích tốt với thiết bị di động.

## BackEnd

### Ngôn ngữ lập trình Java

Java là một ngôn ngữ lập trình hướng đối tượng được giới thiệu bởi Sun Microsystems vào năm 1995. Đặc điểm nổi bật của Java là nguyên tắc "Viết một lần, chạy mọi nơi" (WORA), cho phép lập trình viên viết mã mà không cần phải lo lắng về nền tảng cụ thể, vì mã sẽ được biên dịch thành mã bytecode có thể chạy trên mọi thiết bị có Java Virtual Machine (JVM). Ngôn ngữ này cũng tự động quản lý bộ nhớ thông qua cơ chế thu gom rác, giúp giảm thiểu rủi ro khi lập trình viên phải quản lý bộ nhớ. Hơn nữa, Java cung cấp một bộ thư viện phong phú hỗ trợ cho việc phát triển nhiều loại ứng dụng, từ ứng dụng web và di động (như ứng dụng Android) đến các hệ thống lớn phục vụ cho doanh nghiệp. Nhờ vào cộng đồng lập trình viên đông đảo và các tài nguyên phong phú, Java vẫn giữ được vị trí quan trọng trong lĩnh vực công nghệ thông tin, là sự lựa chọn phổ biến cho nhiều dự án phần mềm hiện nay.

### Thư viện sử dụng

**Spring Data JPA**:

* Spring Data JPA là một phần mở rộng của Spring Data, hỗ trợ tương tác với cơ sở dữ liệu bằng cách cung cấp các API dựa trên JPA (Java Persistence API).
* Nó giúp giảm bớt việc viết mã SQL thủ công, thay vào đó cung cấp các phương thức truy vấn đơn giản với cú pháp rõ ràng và dễ sử dụng.
* Ta có thể tạo các repository để thao tác CRUD (tạo, đọc, cập nhật, xóa) mà không cần viết nhiều mã SQL phức tạp.

**Lombok**:

* Lombok là một thư viện giúp giảm bớt mã boilerplate (mã không cần thiết) trong Java.
* Bằng cách sử dụng các annotation của Lombok như @Getter, @Setter, @Data, và @Builder, ta có thể tự động sinh mã cho các phương thức getter, setter, constructor, builder và nhiều tính năng khác.
* Điều này giúp mã nguồn trở nên ngắn gọn và dễ đọc hơn.

**MapStruct**:

* MapStruct là một thư viện Java dùng để ánh xạ (mapping) giữa các đối tượng DTO và entity một cách tự động.
* Nó giúp chuyển đổi dữ liệu giữa các lớp (classes) một cách nhanh chóng và tránh các lỗi thường gặp trong việc ánh xạ thủ công.
* Bằng cách sử dụng các interface và annotation của MapStruct, bạn có thể định nghĩa các ánh xạ mà không cần viết nhiều mã chuyển đổi.

**Hibernate**:

* Hibernate là một framework ORM (Object-Relational Mapping) phổ biến nhất trong Java, được sử dụng để ánh xạ các lớp Java với các bảng trong cơ sở dữ liệu.
* Nó giúp quản lý các giao dịch, hỗ trợ lazy loading, caching và cung cấp các tính năng như xử lý quan hệ (One-to-Many, Many-to-Many) một cách linh hoạt.
* Hibernate thường được sử dụng cùng với Spring Data JPA, giúp quản lý và truy vấn dữ liệu một cách thuận tiện hơn.

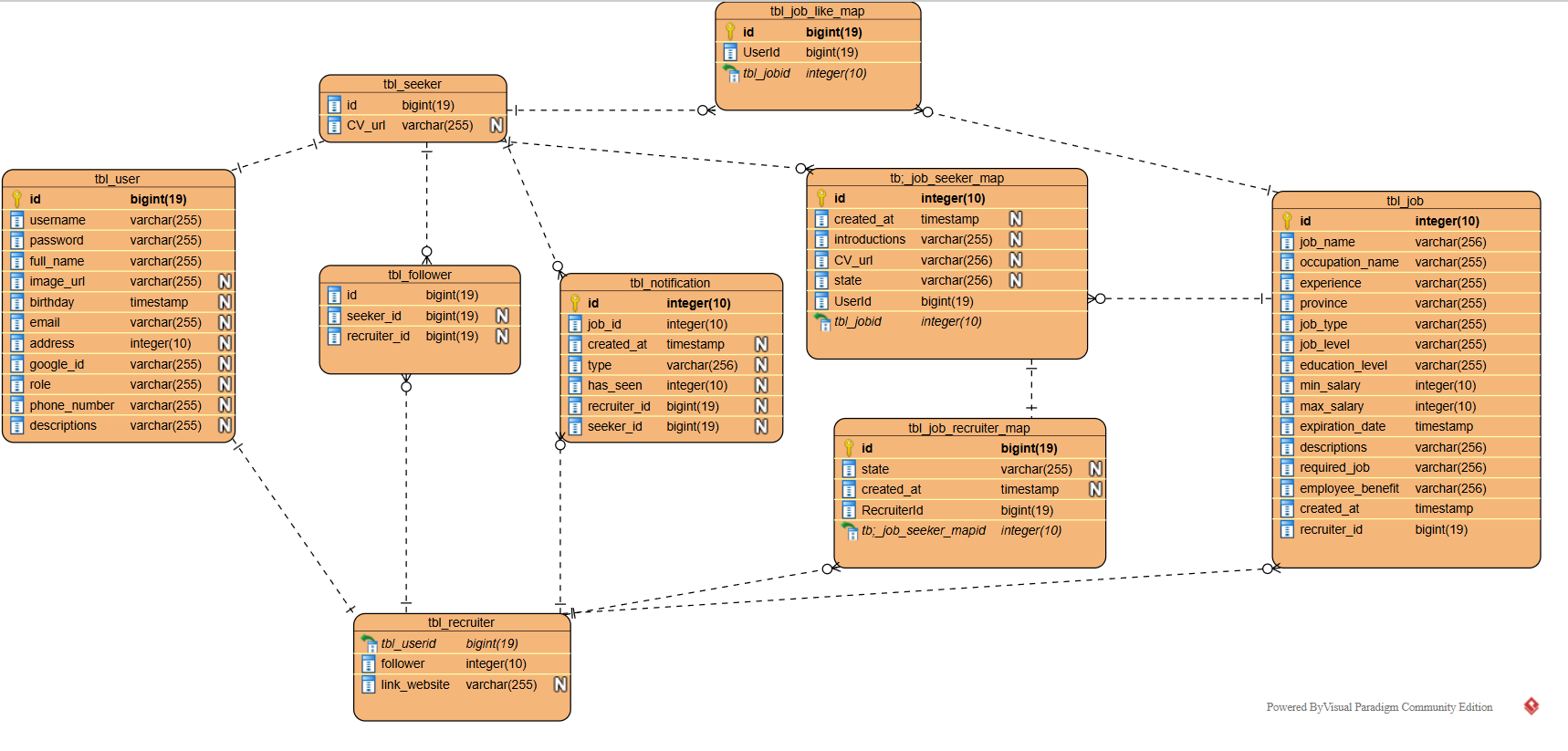
**PostgresSql**

* **PostgresSQL** là một trình điều khiển (driver) JDBC giúp Spring Boot kết nối và tương tác với cơ sở dữ liệu PostgresSQL.
* Khi tích hợp PostgresSQL, ta có thể sử dụng các thư viện như Spring Data JPA và Hibernate để thực hiện các thao tác CRUD, truy vấn dữ liệu, và quản lý giao dịch trên Postgres.
* Driver này giúp chuyển đổi các câu lệnh SQL từ ứng dụng Java thành các lệnh mà Postgres hiểu và thực thi, làm cho việc giao tiếp với cơ sở dữ liệu trở nên liền mạch.

**Cloudinary**:

* Cloudinary là một dịch vụ lưu trữ và quản lý phương tiện truyền thông (media) đám mây, thường được sử dụng để lưu trữ, xử lý và phân phối hình ảnh, video.
* Khi tích hợp Cloudinary, ta có thể tải ảnh và video lên từ ứng dụng của mình và nhận lại URL để truy cập hoặc hiển thị.
* Cloudinary còn cung cấp các API để xử lý ảnh và video, ví dụ như thay đổi kích thước, cắt, nén ảnh, và chuyển đổi định dạng, giúp việc quản lý hình ảnh trong ứng dụng của mình trở nên dễ dàng và hiệu quả.

# THIẾT KẾ CƠ SỞ DỮ LIỆU



Hệ thống tuyển dụng sử dụng cơ sở dữ liệu gồm nhiều bảng để quản lý thông tin người dùng, công việc và các hoạt động tương tác giữa người tìm việc và nhà tuyển dụng. Trung tâm của hệ thống là bảng **tbl\_user**, nơi lưu trữ thông tin đăng nhập và hồ sơ cơ bản của tất cả người dùng. Mỗi người dùng có thể đóng vai trò là **người tìm việc (seeker)** hoặc **nhà tuyển dụng (recruiter)**, và vai trò này được mở rộng qua hai bảng **tbl\_seeker** và **tbl\_recruiter**. Mối quan hệ giữa **tbl\_user** với hai bảng mở rộng này là **một-một**.

Nhà tuyển dụng có thể đăng nhiều công việc, và các công việc này được lưu trong bảng **tbl\_job.** Mỗi bản ghi trong **tbl\_job** chứa thông tin mô tả công việc, yêu cầu ứng viên và mức lương đề xuất. Mối quan hệ giữa **tbl\_recruiter** và **tbl\_job** là **một-nhiều** (một recruiter đăng nhiều job).

Người tìm việc có thể nộp đơn vào nhiều công việc thông qua bảng **tbl\_job\_seeker\_map**. Bảng này đóng vai trò trung gian ghi nhận mối quan hệ giữa người dùng và các công việc mà họ ứng tuyển, kèm theo thời gian nộp, lời giới thiệu và đường dẫn CV. Mỗi bản ghi trong bảng này có thể nhận được phản hồi từ nhà tuyển dụng, và phản hồi đó được lưu trong bảng **tbl\_job\_recruiter\_map**. Bảng này ghi lại trạng thái phản hồi của recruiter như “đã đọc”, “chấp nhận”, “từ chối”... Mối quan hệ từ **tbl\_job\_seeker\_map** đến **tbl\_job\_recruiter\_map** là **một-một**.

Ngoài ứng tuyển, người dùng còn có thể **yêu thích** (like) một công việc thông qua bảng **tbl\_job\_like\_map**. Đây là bảng trung gian thiết lập quan hệ **nhiều-nhiều** giữa **tbl\_user** và **tbl\_job**.

Hệ thống còn hỗ trợ chức năng **theo dõi** giữa recruiter và seeker thông qua bảng **tbl\_follower**. Bảng này ghi nhận những trường hợp recruiter theo dõi seeker, tạo nên mối quan hệ **nhiều-nhiều** giữa hai đối tượng này. Điều này giúp recruiter dễ dàng theo dõi các seeker tiềm năng.

Cuối cùng, để quản lý các tương tác trong hệ thống, bảng **tbl\_notification** được sử dụng để lưu trữ các thông báo giữa recruiter và seeker, như thông báo về trạng thái hồ sơ, lời mời phỏng vấn hoặc phản hồi từ nhà tuyển dụng. Mỗi thông báo có thể liên kết với một công việc cụ thể, người gửi và người nhận, và trạng thái đã xem hay chưa.

Tổng thể, cơ sở dữ liệu được thiết kế xoay quanh người dùng, cho phép tương tác đa chiều giữa seeker và recruiter thông qua ứng tuyển, phản hồi, yêu thích, theo dõi và thông báo. Cách thiết kế này giúp hệ thống linh hoạt và dễ mở rộng, đồng thời hỗ trợ đầy đủ các chức năng cần thiết cho một nền tảng tuyển dụng chuyên nghiệp.

# CHỨC NĂNG

## Đăng ký, đăng nhập

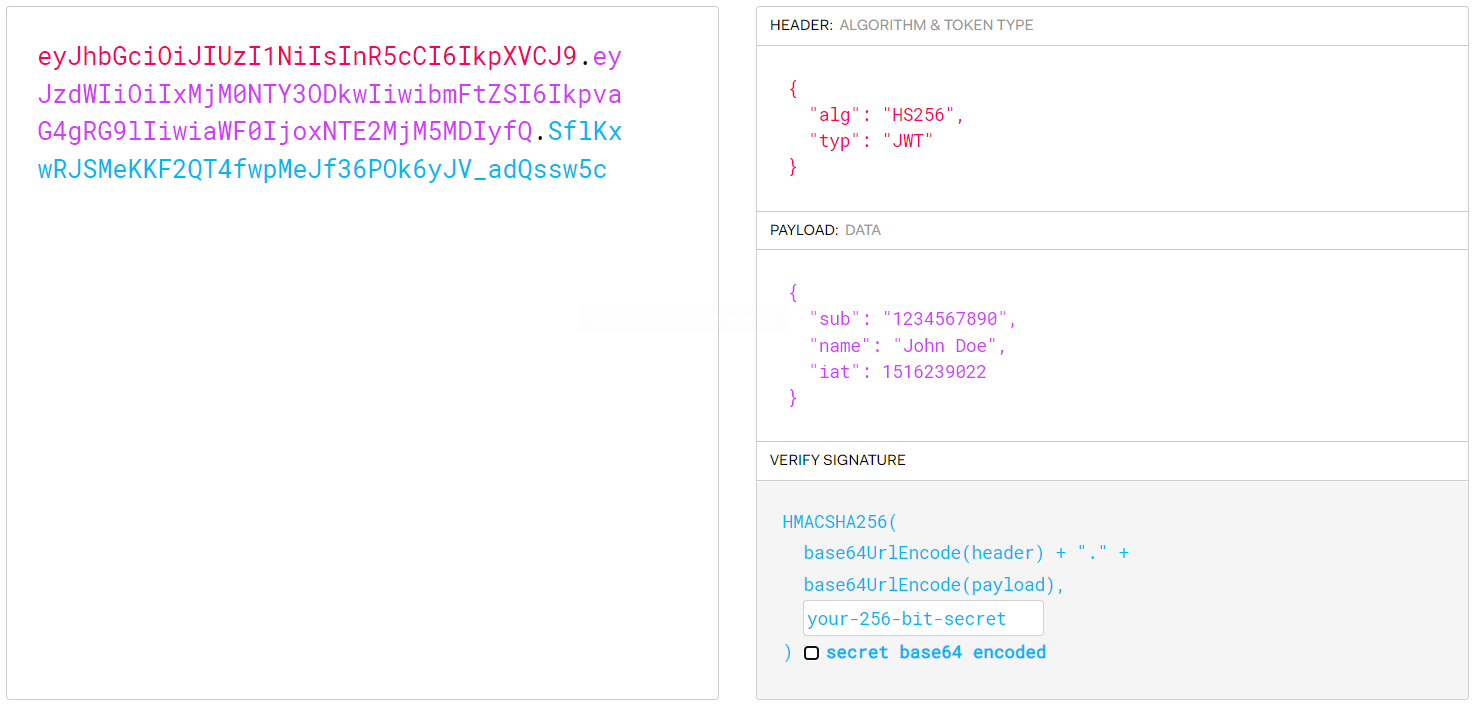
### JSON Web Token

* JSON Web Token là chuỗi mã hóa được sinh ra sau khi người dùng đăng nhập vào ứng dụng, chuỗi mã hóa này được sử dụng để xác minh quyền truy cập vào tài nguyên mà không cần nhập lại username/password nữa.
* Cấu trúc của JWT :
  + Header : gồm 2 phần chính :
    - typ - Loại token (mặc định là JWT - cho biết đây là 1 token JWT)
    - Alg – Là thuật toán dùng để mã hóa.
    - Từ 2 thành phần trên thì header sẽ được mã hóa dựa trên mã hóa

base64url để nhận được đoạn mã bất kì

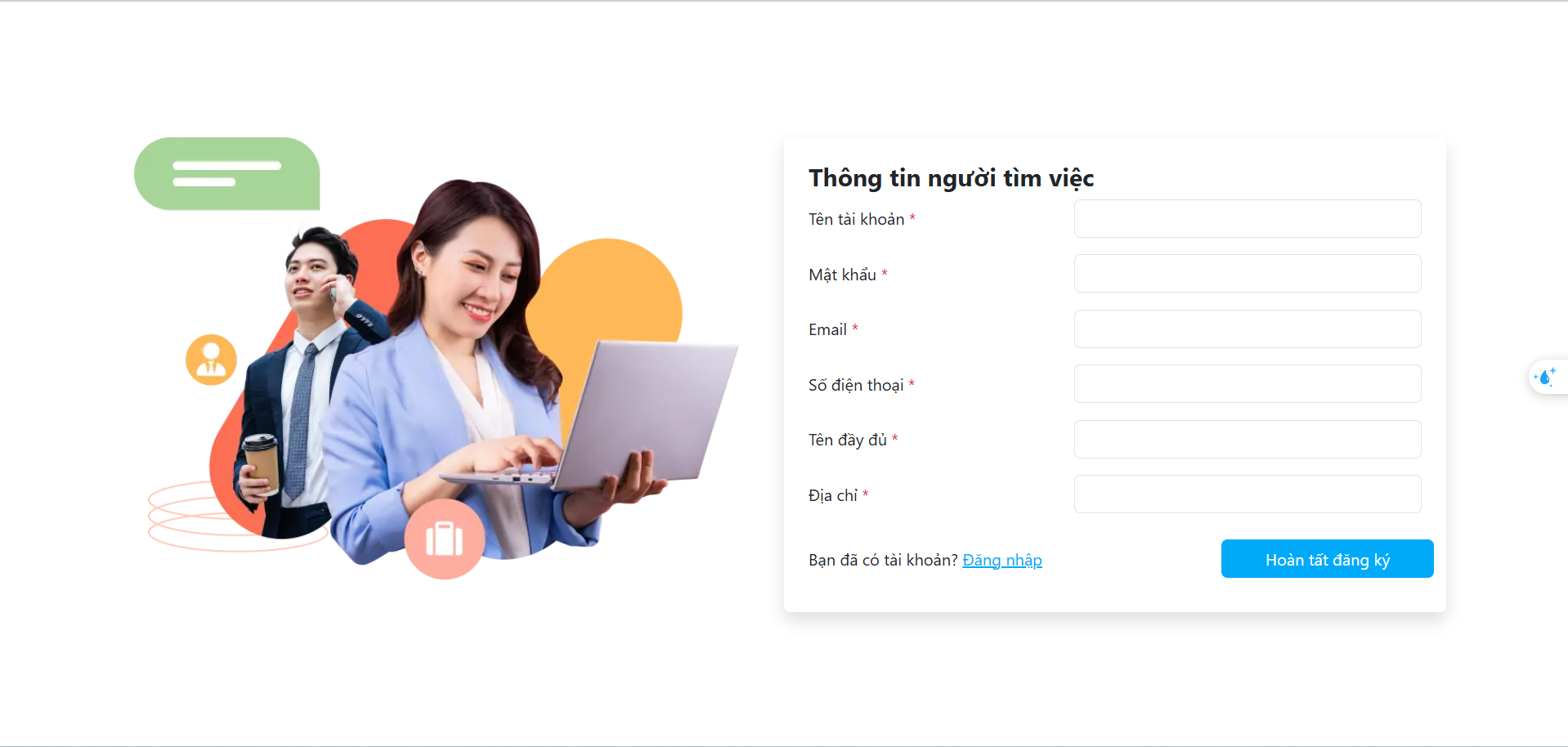
* + Payload : Là nơi chứa các nội dụng thông tin, nó có thể là mô tả 1 thực thể hoặc cũng là cá thông tin bổ sung cho header, VD có thể 1 vài thông tin người dùng hoặc ngày bắt đầu, ngày hết hạn ... Sau đó nó cũng sử dụng mã hóa base64 để nhận được 1 đoạn mã hóa.
* Signature : Là phần kết hợp giữa header và payload đã được mã hóa base64, sau đó nó sẽ kết hợp với 1 cái secretkey và thuật toán mã hóa ở phần header để mã hóa thu được signature cuối cùng. Đây cũng là phần quan trọng nhất để xác định rằng accessToken không bị thay đổi khi truyền tin.
* Sau khi có được 3 thành phần trên, kết hợp chúng và phân tách mỗi phần

bằng dấu chấm.

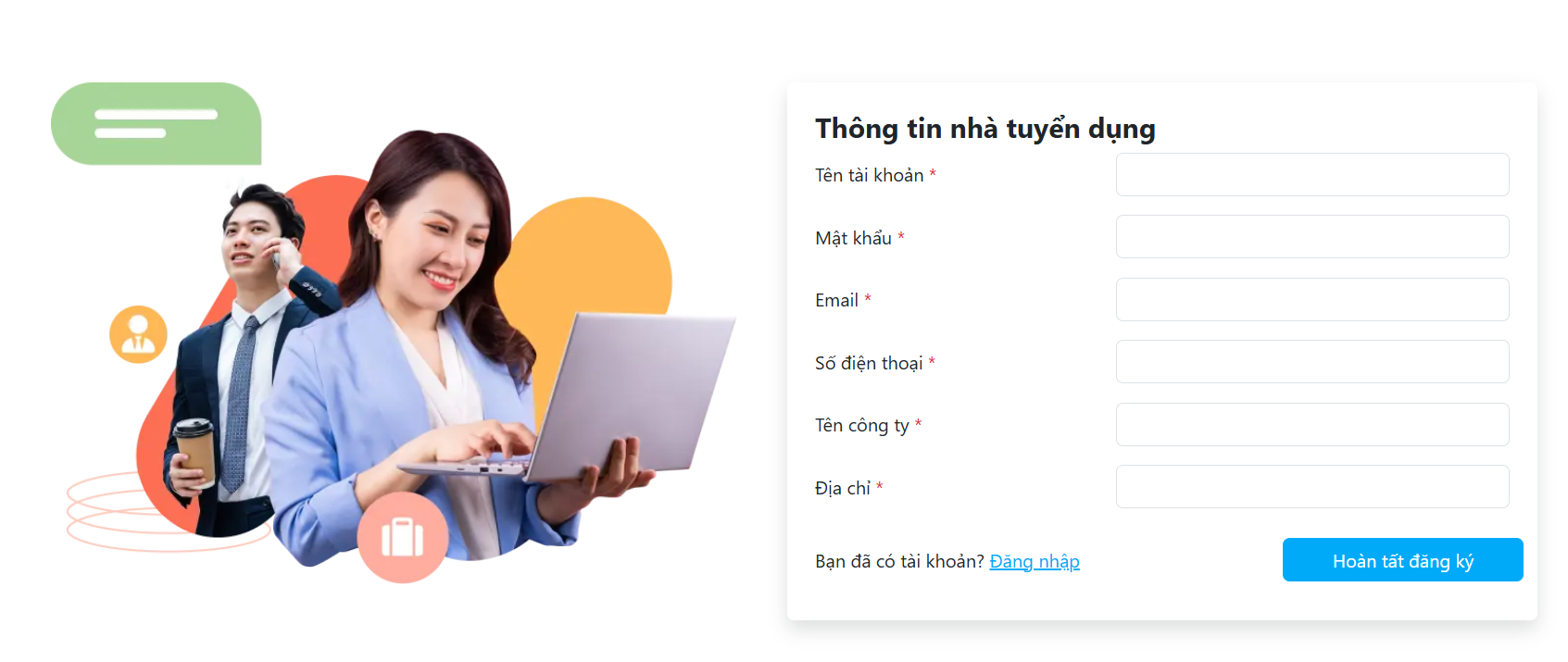


### Đăng ký

* Ở giao diện đăng ký, tùy vào role mà người đăng ký muốn, sẽ hiển thị ra 2 form đăng ký khác nhau ứng với người tìm kiếm việc làm hoặc nhà tuyển dụng.
* Các thông tin ở form đăng ký sẽ được validate như là tên tài khoản, số điện thoại, email đã tồn tại trong cơ sở dữ liệu hay chưa, nếu đã tồn tại sẽ trả về thông báo cho người dùng.
* Sau khi đăng kí thành công, người dùng sẽ được điều hướng trở lại trang đăng nhập của hệ thống.

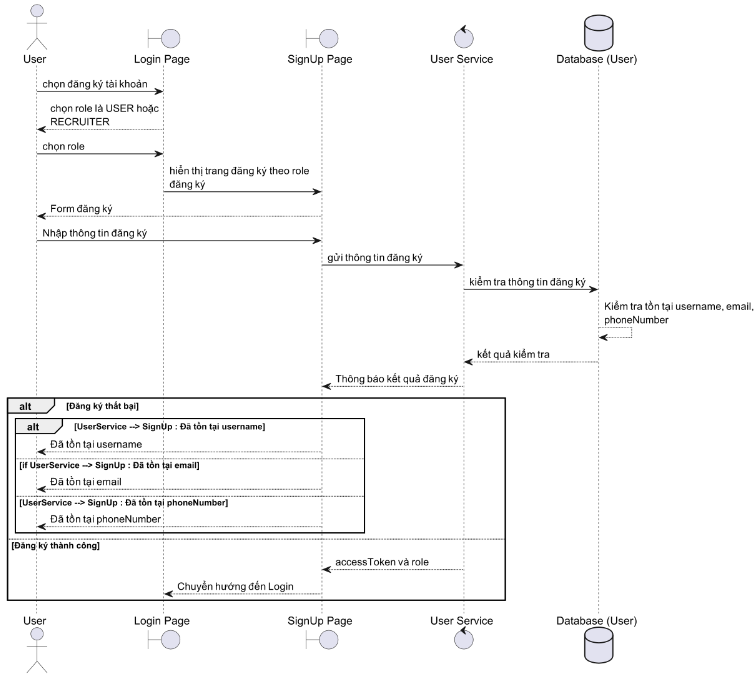


#### Form đăng ký dành cho người tìm việc

****

#### Form đăng ký dành cho nhà tuyển dụng

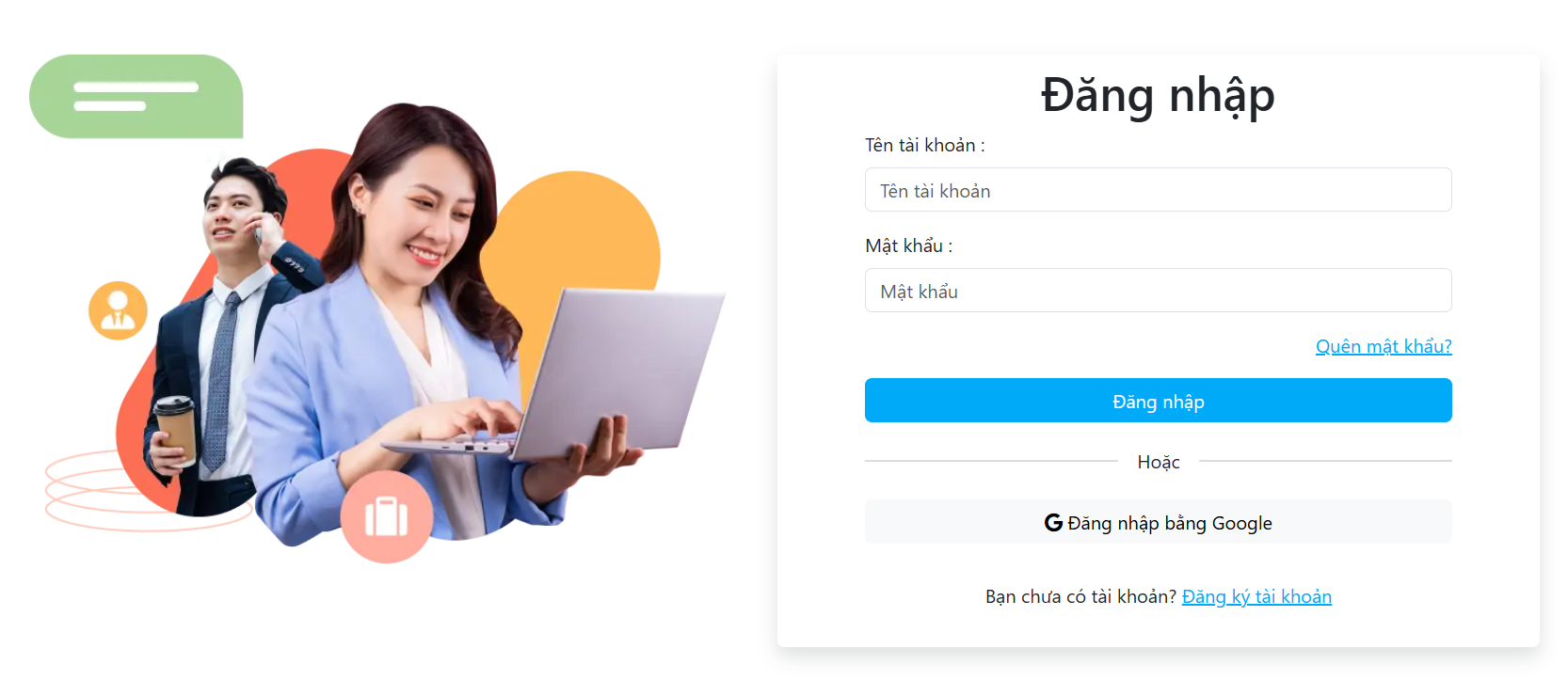
* Tùy thuộc vào role mà người dùng chọn, sẽ cần điền các thông tin khác nhau.
* Dưới đây là sơ đồ luồng hoạt động của việc đăng ký tài khoản :



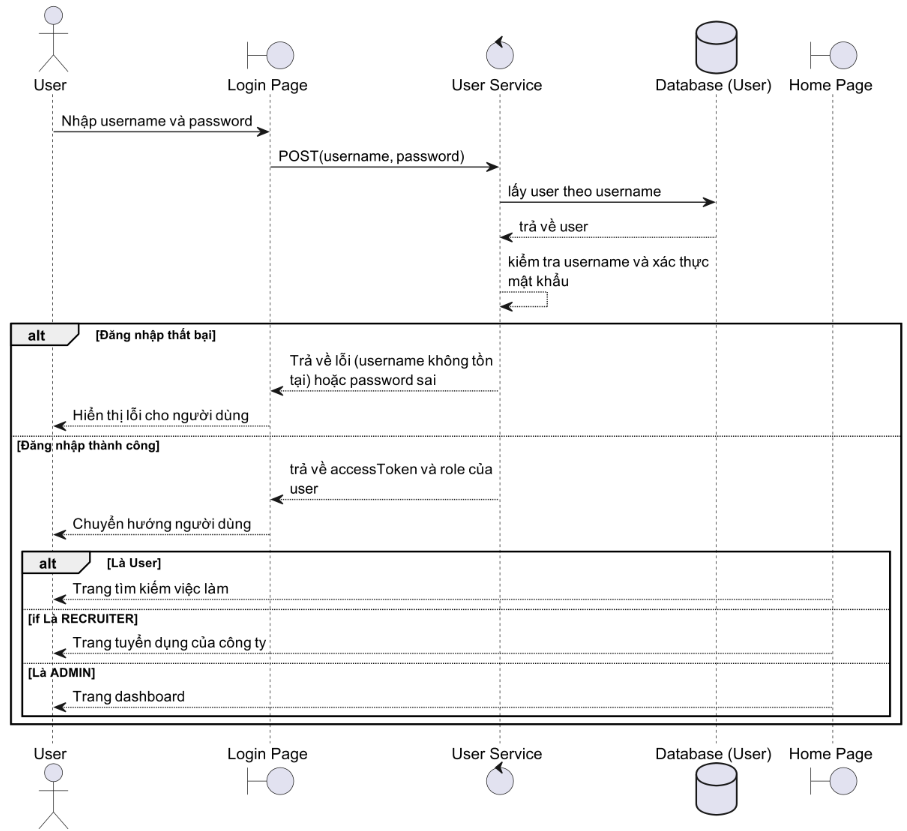
#### Luồng đăng ký tài khoản

### Đăng nhập sử dụng username/password

* Sau khi đăng ký thành công, người dùng sẽ nhập tài khoản, mật khẩu đã đăng ký từ trước đó để đăng nhập.
* Tại bước đăng nhập, sau khi gửi thông tin đăng nhập, phía server sẽ kiểm tra xem tên tài khoản đã tồn tại hay chưa, đồng thời kiểm tra mật khẩu đã đúng hay chưa.
* Nếu tên tài khoản chưa tồn tại hoặc sai mật khẩu sẽ thông báo trả lại người dùng. Nếu tên tài khoản và mật khẩu đã chính xác, người dùng sẽ được điều hướng đến giao diện ứng với role của người dùng đã đăng ký từ trước đó.
* Đồng thời phía FE cũng sẽ nhận được accessToken và role của người dùng đó. AccessToken được lưu lại và sử dụng cho các yêu cầu sau này của người dùng.
* Dưới đây là hình ảnh giao diện đăng nhập cũng như là luồng hoạt động của chức năng đăng nhập.



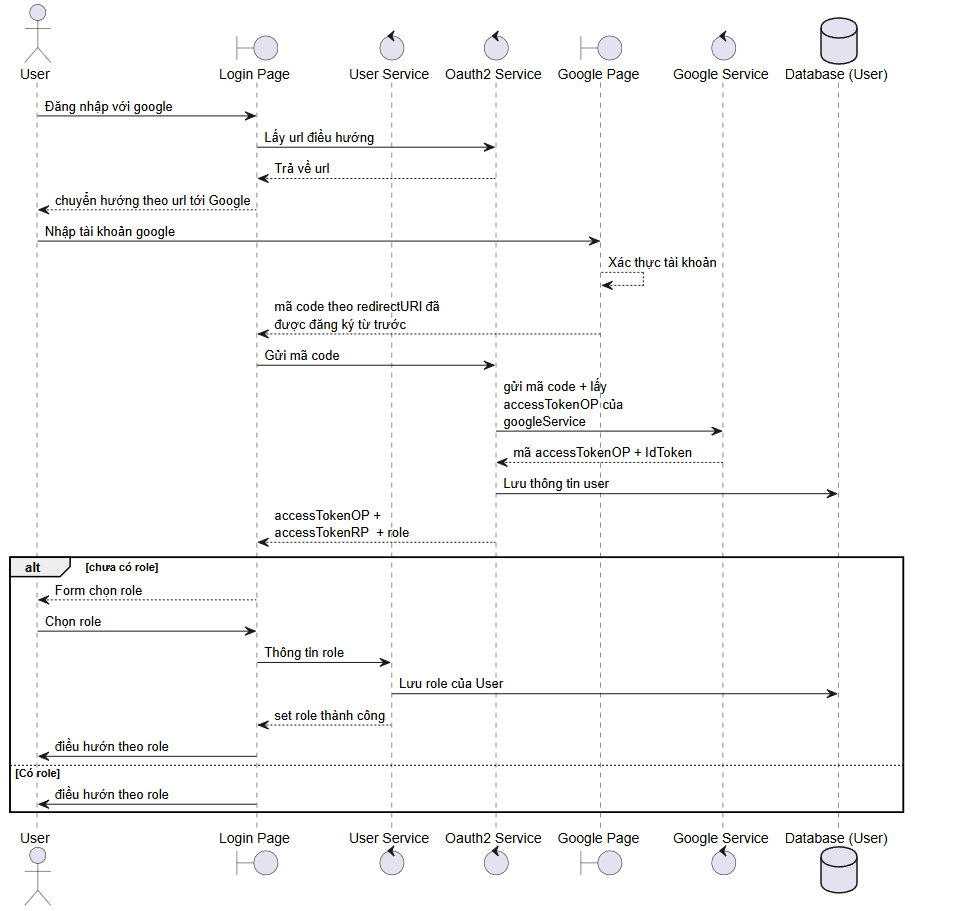
#### Giao diện đăng nhập bằng usename/password



#### Luồng hoạt động của chức năng đăng nhập bằng username/password

### Đăng nhập bằng google

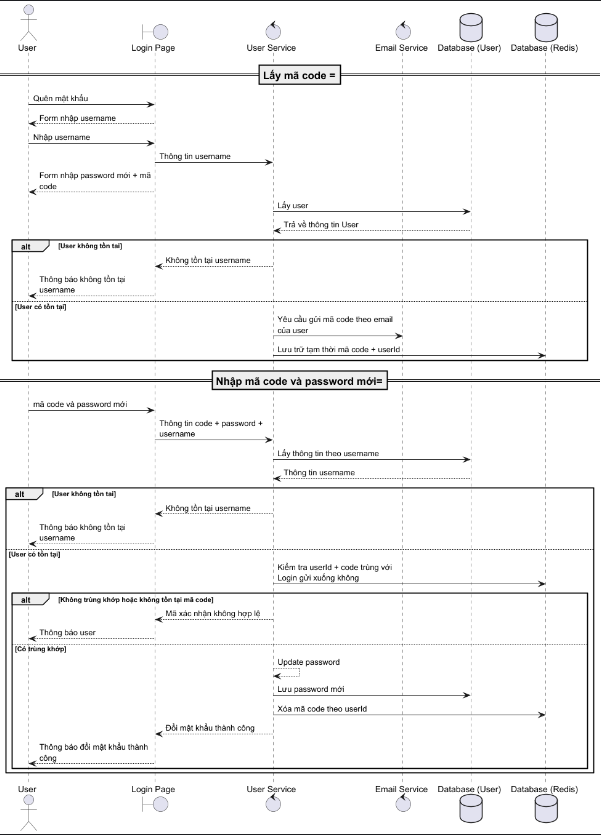
* Nếu người dùng không muốn đăng nhập bằng username/password họ có thể sử dụng đăng nhập bằng tài khoản google.
* Khi đăng nhập bằng google, phía Front End sẽ gọi tới server Backend để lấy url điều hướng, sau khi nhận được url phía Front End sẽ chuyển người dùng tới trang đăng nhập google.
* Sau khi chọn tài khoản đăng nhập, phía google sẽ trả về 1 mã code trên url\_redirect, phía Front End sẽ lấy mã code đó để truyền tới server Backend.
* Sau khi nhận được mã code, server bắt đầu gửi lại mã code kèm với yêu cầu cấp accessToken và idToken từ phía nhà cung cấp google.
* Google sau khi nhận được yêu cầu sẽ xác minh mã code cũng như danh tính server, nếu hợp lệ nó sẽ trả về accessToken và idToken có thể là cả refreshToken.
* Sau khi nhận được accessToken, server Backend kiểm tra xem người dùng đó đã tồn tại trong cơ sở dữ liệu chưa, nếu chưa sẽ tạo mới người dùng và trả về phía Front End accessToken và role.
* Phía Front End sẽ kiểm tra xem role có tồn tại hay không, nếu có thì chuyển tới trang ứng với role, nếu không thì sẽ cho người dùng chọn role rồi mới chuyển hướng tới trang.
* Dưới đây là luồng đăng ký/đăng nhập bằng google :



#### Đăng ký/ đăng nhập bằng google

### Quên mật khẩu

* Khi người dùng chọn quên mật khẩu, Front End sẽ hiển thị ra 1 form cho phép người dùng nhập username.
* Sau khi nhập username, nó sẽ gọi tới server Backend để gửi 1 mã code đến email tương ứng với username và lưu lại mã code đó để đối chiếu.
* Sau đó, người dùng nhập mã code + mật khẩu mới và gửi tới server Backend, server sẽ kiểm tra mã code đó có hợp lệ hay không, nếu hợp lệ sẽ thay đổi mật khẩu mới.
* Dưới đây là luồng hoạt động :



#### Quên mật khẩu